MATHEMATICAL ENGINEERING TECHNICAL REPORTS

Cut Sparsifiers for Balanced Digraphs

Motoki IKEDA and Shin-ichi TANIGAWA

METR 2018–07 August 2018

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: https://www.keisu.t.u-tokyo.ac.jp/research/techrep/

Cut Sparsifiers for Balanced Digraphs*

Motoki Ikeda and Shin-ichi Tanigawa

Department of Mathematical Informatics
Graduate School of Information Science and Technology
The University of Tokyo
{motoki_ikeda,tanigawa}@mist.i.u-tokyo.ac.jp

August, 2018

Abstract

In this paper we consider a cut sparsification problem for digraphs parametrized by balancedness. A weighted digraph $D = (V, E)$ is said to be α-balanced if the total weight of the edges from U to $V \setminus U$ is at most α times the total weight of the edges from $V \setminus U$ to U for any $U \subseteq V$. Based on the combinatorial cut-sparsification framework by Fung et al. (2011), we show that for any α-balanced weighted digraph D with n vertices and m edges there is a weighted subdigraph D' with $O(\alpha\epsilon^{-2} n \log n \log(nW))$ edges that $(1+\epsilon)$-cut-approximates D where W is the maximum weight of an edge in D. We also show how to compute a such cut sparsifier in $O(m \log \alpha + \alpha^3 n \log W \text{poly}(\log n))$ time.

Applying our sparsifier as a preprocessing, the running time of the minimum cut approximation algorithm by Ene et al. (2016) is improved to $O(m \log \alpha + \alpha^3 \epsilon^{-4} n \text{poly}(\log n))$ for an α-balanced digraph with n vertices and m edges.

1 Introduction

Graph sparsification is one of the fundamental tools for developing efficient graph algorithms. The seminal work of Karger [9] and Benczúr and Karger [1, 2] showed that for any positively weighted undirected graph G with n vertices and m edges, there is a weighted subgraph G' with $O(\epsilon^{-2} n \log n)$ edges such that the size of each cut is within $(1 \pm \epsilon)$ factor of the original cut size. Such a sparse subgraph is called a *cut sparsifier*. They also gave an $O(m \log^3 n)$ time algorithm for constructing a cut sparsifier with high probability, and demonstrated applications to several cut and flow problems. Later, Spielman and Teng [15] introduced a generalized notion, a *spectral sparsifier*, that sparsifies G keeping the spectral of the Laplacian, and have broadened applications to solving linear systems. Since the work of [15], various improved spectral sparsifiers and efficient algorithms have been developed.

This successful line of research is only for undirected graphs, and despite its obvious importance, there has been little progress for digraphs. Cohen et al. [4] recently introduced a new notion of spectral sparsifiers based on a scaled norm, and they showed the existence of sparsifiers with $O(\epsilon^{-2} n \text{poly}(\log n))$ edges for any strongly connected digraphs. However, unlike the undirected case, their spectral sparsifier does not imply a cut sparsifier. In fact there are digraphs which do not admit cut sparsifiers with sub-quadratic size (see [4]). This is a typical reason why there is no counterpart theory for digraphs.

Although we cannot hope for a perfect theory for digraphs, there is a natural question; for which class of digraphs can we construct good cut sparsifiers? In this paper we study this problem

*This work is supported by JST CREST (JPMJCR1402).

by focusing on balanced graphs. Balancedness is a new notion introduced by Ene et al. [5] for expressing the ratio of the in-coming and out-going cut sizes. More formally, for $\alpha \geq 1$, a digraph $D = (V, E)$ is called *α-balanced* if

$$\delta^+(U; D) \leq \alpha \delta^-(U; D)$$

holds for any $U \subseteq V$, where $\delta^+(U; D)$ (resp., $\delta^-(U; D)$) denotes the sum of the weights of the edges from U to $V \setminus U$ (resp., from $V \setminus U$ to U). The *imbalance* b_D of D is defined to be the infimum of α such that D is α-balanced. Note that $b_D = 1$ if and only if D is Eulerian.

The main contribution of this paper is to show the existence of cut sparsifiers whose sizes are parametrized by b_D. We show that for any weighted digraph D with n vertices and m edges, there is a weighted subdigraph D' with $O(b_D \epsilon^{-2} n \log n \log(nW))$ edges such that

$$(1 - \epsilon)\delta^+(U; D) \leq \delta^+(U; D') \leq (1 + \epsilon)\delta^+(U; D) \quad \text{for all } U \subseteq V,$$

where W is the maximum weight of an edge in D. We further show how to obtain such a cut sparsifier in $O(m \log b_D + b_D^3 n \log W \mathrm{poly}(\log n))$ time with high probability.

Our result on the existence of cut sparsifiers is actually a direct application of a result on undirected cut sparsifiers. Although the main focus of the research for undirected graphs has been shifted to spectral sparsifiers, still interesting questions remain even for cut sparsifiers. One such a question is to understand which graph parameter can be used as a sampling parameter in a sampling-type algorithm. Fung et al. [6] gave a general framework to solve this question for undirected graphs. In this paper we exploit the power of their remarkable combinatorial approach; we show that the proof of the main result in [6] can be applied even to digraphs without any substantial modification.

As is always the case with cut sparsifiers, our result can be used as a preprocessing of algorithms for any cut problem. One interesting example is the minimum cut problem of balanced digraphs studied by Ene et al. [5]. Ene et al. [5] gave an algorithm to find a $(1+\epsilon)$-approximate minimum cut (and a $(1-\epsilon)$-approximate maximum flow) of a digraph D that runs in $O(m b_D^2 \epsilon^{-2} \log^c n)$ time for some constant c. (Here the current best c is 45, see [14].) Using our sparsifier at a preprocessing phase, we obtain an algorithm that runs in $O(m \log b_D + b_D^3 \epsilon^{-4} n \,\mathrm{poly}(\log n))$ time. This is a substantial improvement if b_D is not too large. (Note that an exact algorithm in [11] is faster than that of Ene et al. [5] if $b_D = \Omega(n^{1/4})$.)

The paper is organized as follows. In Section 2 we show the existence of cut sparsifiers for balanced digraphs, and in Section 3 we give an efficient algorithm for constructing those sparsifiers. In Section 4 we explain an application to the minimum cut problem. In Section 5 we give a short remark on the number of cut projections in α-balanced digraphs.

Throughout the paper we consider a digraph $D = (V, E)$ or an undirected graph $G = (V, E)$ with n vertices, m edges, and each edge weight is a positive integer. As defined above, for $U \subseteq V$, $\delta^+(U; D)$ (resp., $\delta^-(U; D)$) denotes the sum of the weights of the edges from U to $V \setminus U$ (resp., from $V \setminus U$ to U). In an undirected graph G, we use $\delta(U; G)$ to denote the sum of the weights of the edges between U and $V \setminus U$. The (local) *edge connectivity* $\kappa(e; G)$ of $e = \{u, v\}$ in G is defined by $\kappa(e; G) = \kappa_e = \min\{\delta(U; G) \mid U \subseteq V, u \in U, v \notin U\}$. Similarly, the edge connectivity $\kappa(e; D)$ of $e = (u, v)$ in D is defined by $\kappa(e; D) = \min\{\delta^+(U; D) \mid U \subseteq V, u \in U, v \notin U\}$.

2 Digraph Sparsification

In this section, we give cut sparsifications for digraphs based on the result by Fung et al. [6]. Let us first give the following formal definition.

Definition 2.1. *Let $D = (V, E)$ be a digraph. A digraph $D' = (V, E')$ ϵ-*cut-approximates *D, which is often abbreviated as $D' \in (1 \pm \epsilon)D$, if for all $U \subseteq V$,*

$$(1 - \epsilon)\delta^+(U; D) \leq \delta^+(U; D') \leq (1 + \epsilon)\delta^+(U; D).$$

A sparse subgraph that ϵ-cut-approximates the original graph is called a cut sparsifier.

As is in the ordinary sparsification framework, our algorithm is a random sampling algorithm. More specifically, we use the *compression* of each edge, first introduced by Benczúr and Karger [1, 2], where each edge e is sampled with probability p_e and the sampled edge is given a weight $1/p_e$. The sampling probability is determined by a graph parameter λ_e for each edge e. The original algorithm by Benczúr and Karger [1, 2] uses the strong connectivity of each edge e for λ_e, which is defined to be the largest k for which a k-edge-connected subgraph containing the edge exists. Fung et al. [6] showed that it is possible to construct a cut sparsifier using edge connectivity, effective resistance, or Nagamochi-Ibaraki index (defined in Section 2.1).

A formal description of the *compression* for digraphs is given in Algorithm 1.

Algorithm 1 Compress$(D, \lambda, \gamma, d, \epsilon)$

Input: A weighted simple digraph $D = (V, E, w)$ with weight $\omega : E \to \mathbb{Z}_+$, an edge parameter $\lambda : E \to \mathbb{Z}_+$, a constant $\gamma, d \in \mathbb{R}_+$, and $\epsilon \in (0, 1)$

Output: A cut sparsifier $D_\epsilon = (V, F, u)$

$C \leftarrow 43(d + 7)$
$\rho \leftarrow C\gamma \ln n/\epsilon^2$
$F \leftarrow \emptyset$
for each $e \in E$ **do**
  $p_e \leftarrow \min\{\rho/\lambda_e, 1\}$
  Generate a random number X_e from a binomial distribution $B(w_e, p_e)$
  if $X_e > 0$ **then**
    Add edge e to F and set $u_e = X_e/p_e$
  end if
end for
return $D_\epsilon = (V, F, u)$

We now analyze the quality of the output D_ϵ. Following the analysis by Fung et al. [6], we consider a partition $F_0, F_1, \ldots, F_\Lambda$ of the edge set E of D defined by

$$F_i := \{e \in E \mid 2^i \leq \lambda_e < 2^{i+1}\}$$

where $\Lambda = \lfloor \lg(\max_{e \in E} \lambda_e) \rfloor$.

We say that a family $G_0, \ldots, G_\Lambda$ of weighted undirected graphs *covers* D if for each i and for each $(u, v) \in F_i$, the weight of $\{u, v\}$ in G_i is greater than or equal to the sum of the weights of (u, v) and (v, u) in F_i. Such a cover is said to be a γ-*certificate* [1] if the following two properties are satisfied:

(Connectivity) For each $i \geq 0$ and each edge $(u, v) \in F_i$, $\kappa(\{u, v\}; G_i) \geq 2^{i-1}$.

(Overlapped) For any $U \subseteq V$, $\sum_{i=0}^{\Lambda} \delta(U; G_i) \leq \gamma \cdot \delta^+(U; D)$.

Given γ-certificates, the following theorem states the existence of cut sparsifiers.

Theorem 2.2. *Let D be a weighted digraph, and λ_e be a positive integer for each $e \in E$. Suppose that there exists a γ-certificate family of weighted undirected graphs that covers D. Then, $D_\epsilon =$ Compress$(D, \lambda, \gamma, d, \epsilon)$ contains $O(\frac{\gamma \log n}{\epsilon^2} \sum_{e \in E} \frac{w_e}{\lambda_e})$ edges in expectation, and $D_\epsilon \in (1 \pm \epsilon)D$ with probability at least $1 - 1/n^d$.*

[1] This is a simplified and adapted notion of the (π, α)-certificate introduced by Fung et al. [6].

Proof. The theorem follows from the following more general statement, Theorem 2.3, by observing that each undirected graph G_i is considered as an Eulerian digraph if we regard each undirected edge as two parallel directed edges of both directions. $\square$

We can apply the above definition of a covering family and a γ-certificate to a family of weighted digraphs $D_0, \ldots, D_\Lambda$. Formally, a family $D_0, \ldots, D_\Lambda$ of weighted digraphs *covers* D if for each i and for each $(u, v) \in F_i$, the weight of (u, v) in D_i is greater than or equal to the weight of (u, v) in F_i. A cover is said to be a γ-*certificate* if $\kappa((u, v); D_i) \geq 2^{i-1}$ holds for each $i \geq 0$ and each edge $(u, v) \in F_i$, and $\sum_{i=0}^{\Lambda} \delta^+(U; D_i) \leq \gamma \cdot \delta^+(U; D)$ for any $U \subseteq V$.

Theorem 2.2 still holds if G_i is substituted by an Eulerian digraph D_i.

Theorem 2.3. *Let D be a weighted digraph, and λ_e be a positive integer for each $e \in E$. Suppose that there exists a γ-certificate family of weighted Eulerian digraphs that covers D. Then, $D_\epsilon = \mathsf{Compress}(D, \lambda, \gamma, d, \epsilon)$ contains $O(\frac{\gamma \log n}{\epsilon^2} \sum_{e \in E} \frac{w_e}{\lambda_e})$ edges in expectation, and $D_\epsilon \in (1 \pm \epsilon)D$ with probability at least $1 - 1/n^d$.*

Theorem 2.3 is a proper extension of Theorem 2.2. The proof of Theorem 2.3 is an adaptation of that of Fung et al. [6], but for completeness we give a formal proof in Appendix B.

2.1 Compression using NI Indexes

In the following two subsections, we shall show how to set up parameter λ_e to apply Theorem 2.2.

Nagamochi and Ibaraki [13, 12] showed how to compute a sparse certificate for the k-connectivity of undirected graphs. Motivated by their work, Fung et al. [6] introduced the following simplified variant of the local connectivity.

Definition 2.4 (NI forest, NI index [6]). *Let G be an undirected graph with integer-valued edge weight, and let $\tilde{G}$ be the multigraph obtained from G by replacing each edge e with weight w_e by w_e parallel edges. A sequence of edge-disjoint spanning forests $T_1, T_2, \ldots$ of $\tilde{G}$ is said to be an* NI forest *packing if T_i is a spanning forest on the edges left in $\tilde{G}$ after removing those in $T_1, T_2, \ldots, T_{i-1}$. An edge with weight w_e in G must appear in w_e contiguous forests. The* NI index *of edge e in G, denoted ℓ_e, is the index of the last NI forest in which e appears.*

Let $D = (V, E)$ be an α-balanced digraph, and G be the undirected graph obtained from D by ignoring the direction. For each edge $e \in E$, we set $\lambda_e = \ell_e$, where ℓ_e is the NI index of e in G. It turns out that the compression using this parameter gives a good sparsifier. To see this we need to construct a family $G_0, \ldots, G_\Lambda$ of undirected graphs with the properties as given in Theorem 2.2.

Let $T_1, T_2, \ldots, T_k$ be an NI forest packing of G. We define a weighted undirected graph H_i to be the union of $T_{2^{i-1}}, T_{2^{i-1}+1}, \ldots, T_{2^i-1}$ (i.e., the weight of $\{u, v\}$ is the number of appearances of edge $\{u, v\}$ in $T_{2^{i-1}}, T_{2^{i-1}+1}, \ldots, T_{2^i-1}$.) We then define $G_i = (V, E_i)$ such that the weight of $\{u, v\}$ is the sum of the weight of $\{u, v\}$ in H_i and the weights of (u, v) and (v, u) in F_i for every pair $u, v \in V$ (and E_i is defined to be the set of pairs of vertices with nonzero weight).

Lemma 2.5. *A family G_i of undirected graphs defined above is a $2(1 + \alpha)$-certificate covering D.*

Proof. Clearly the family covers D.

To see the connectivity, recall first that $\lambda_e \geq 2^i$ for any $e = (u, v) \in F_i$. Hence u and v are connected in each of $T_{2^{i-1}}, T_{2^{i-1}+1}, \ldots, T_{2^i-1}$ by the definition of NI forest packing. Therefore $\kappa(\{u, v\}; G_i) \geq 2^{i-1}$.

To evaluate the overlapping, note that for any $i \neq j$, $F_i \cap F_j = \emptyset$ and the edge set of H_i is disjoint from that of H_j. Hence the sum of the weights of $\{u, v\}$ over G_i is at most two times the weight of $\{u, v\}$ in G. Thus for each $U \subseteq V$ we get $\sum_i \delta(U; G_i) \leq 2\delta(U; G) \leq 2(1 + \alpha)\delta^+(U, D)$, and it is $2(1 + \alpha)$-overlapped. $\square$

We can now apply Theorem 2.2.

Theorem 2.6. *Let D be a weighted digraph, and $D_\epsilon = \mathsf{Compress}(D, \ell, 2(1 + b_D), d, \epsilon)$. Then, D_ϵ contains $O(b_D \epsilon^{-2} n \log n \log(nW))$ edges in expectation, and $D_\epsilon \in (1 \pm \epsilon)D$ with probability at least $1 - 1/n^d$, where W is the maximum weight of an edge in D.*

Proof. By Lemma 2.5, there always exists a $2(1+b_D)$-certificate covering D. Thus by Theorem 2.2, we have a weighted subgraph D_ϵ with $O(\rho \sum_e w_e/\ell_e)$ edges and $D_\epsilon \in (1 \pm \epsilon)D$ with probability at least $1 - 1/n^d$. It was shown by Fung et al. [6] that $\sum_{e \in E} w_e/\ell_e = O(n \log(nW))$. Therefore D_ϵ has the properties in the statement. □

2.2 Compression using Edge Connectivities

If we use the local edge connectivity, we have a slightly better sparsifier. But computing the local edge connectivities is more expensive than computing the NI indexes.

Let $D = (V, E)$ be a digraph, and let G be the undirected graph obtained from D by ignoring the direction. For an edge $e = (u, v)$ in D, we consider the local edge connectivity κ_e of $\{u, v\}$ in G. We consider the compression by setting $\lambda_e = \kappa_e$ for each $e \in E$. We need to construct a family $G_0, \dots, G_\Lambda$ of undirected graphs with the properties as given in Theorem 2.2.

Let $T_1, \dots, T_k$ be an NI forest packing of G. We define a weighted undirected graph H_i to be the union of $T_1, T_2, \dots, T_{2^{i-1}-1}$ for $i \leq \lg n$, the union of $T_{2^{i-1-\lg n}}, T_{2^{i-1-\lg n}+1}, \dots, T_{2^{i+1}-1}$ for $i \geq \lg n + 1$. We then define $G_i = (V, E_i)$ such that the weight of $\{u, v\}$ is the sum of the weight of $\{u, v\}$ in H_i and the weights of (u, v) and (v, u) in F_i for every pair $u, v \in V$ (and E_i is defined to be the set of pairs of vertices with nonzero weight).

Lemma 2.7 (Fung et al. [6])**.** *Let $T_1, T_2, \dots$ be an NI forest packing of an undirected graph $G = (V, E)$. For any pair of vertices $u, v \in V$ and for any $i \geq 1$, $\kappa(u, v; T_1 \cup T_2 \cup \dots \cup T_i) \geq \min\{\kappa_{uv}, i\}$.*

Lemma 2.8. *A family G_i of undirected graphs defined above is a $(1 + \alpha)(3 + \lg n)$-certificate covering D.*

Proof. Clearly the family covers D.

Hence, for $i \leq \lg n$, it holds that $\kappa(\{u, v\}; H_i) \geq 2^{i-1} - 1$ by Lemma 2.7, and $\kappa(\{u, v\}; G_i) \geq 2^{i-1}$. For $i \geq \lg n + 1$, it holds that $\kappa(\{u, v\}; T_1 \cup \dots \cup T_{2^{i+1}-1}) \geq 2^i$ by Lemma 2.7. Since there are at most 2^{i-1} edges in $T_1, T_2, \dots, T_{2^{i-1-\lg n}-1}$, we have $\kappa(\{u, v\}; G_i) \geq 2^{i-1}$.

To evaluate the overlapping, note that for any $i \neq j$, $F_i \cap F_j = \emptyset$ and each edge of G appears in H_i for at most $2 + \lg n$ different values of i. Hence the sum of the weights of $\{u, v\}$ over G_i is at most $3 + \lg n$ times the weight of $\{u, v\}$ in G. Thus for each $U \subseteq V$ we get $\sum_i \delta(U; G_i) \leq (3 + \lg n)\delta(U; G) \leq (1 + \alpha)(3 + \lg n)\delta^+(U; D)$, and it is $(1 + \alpha)(3 + \lg n)$-overlapped. □

We can now apply Theorem 2.2.

Theorem 2.9. *Let D be a weighted digraph, and $D_\epsilon = \mathsf{Compress}(D, \kappa, (1 + b_D)(3 + \lg n), d, \epsilon)$. Then, D_ϵ contains $O(b_D \epsilon^{-2} n \log^2 n)$ edges in expectation, and $D_\epsilon \in (1 \pm \epsilon)D$ with probability at least $1 - 1/n^d$.*

Proof. By Lemma 2.8, there always exists a $(1 + b_D)(3 + \lg n)$-certificate covering D. Thus by Theorem 2.2, we have a weighted subgraph D_ϵ with $O(\rho \sum_e w_e/\kappa_e)$ edges and $D_\epsilon \in (1 \pm \epsilon)D$ with probability at least $1 - 1/n^d$. It is known [6] that $\sum_{e \in E} w_e/\kappa_e \leq n - 1$. Therefore D_ϵ has the properties in the statement. □

We can also apply the analysis to the compression algorithm using the local edge connectivity of digraphs (rather than that of the underlying undirected graphs) as sampling parameter λ_e. However, the resulting edge density is no better than that in Theorem 2.9.

3 Digraph Sparsification Algorithm

In this section we give an efficient implementation of Compress based on the NI index. For this, we compute the NI index of a weighted graph before calling Compress. It is implicit in the work by Nagamochi and Ibaraki [13] that the NI index of a weighted graph can be computed in $O(m + n\log n)$ time. The generation of a random variable from a binomial distribution $B(w_e, p_e)$ can be done in $O(w_e p_e)$ time (see e.g. [8]). Therefore, $\mathsf{Compress}(D, \ell, 2(1+\alpha), \epsilon)$ can be implemented in $O(m + \sum_e w_e p_e)$ time if we know that D is α-balanced in advance. Here $\sum_e w_e p_e = O(\alpha\epsilon^{-2} n \log n \log(nW))$ is the expected number of the edges in the sparsifier, and we may always assume that it is $O(m)$ since otherwise we can simply return D as a better sparsifier. Hence the total running time is $O(m)$. To apply the algorithm to any digraph D, we need to (approximately) compute the imbalance of D. For this, the following result is known.

Lemma 3.1 (Ene et al. [5, Lemma 2.9])**.** *Given a weighted digraph D and α such that D is α-balanced, there is an algorithm $\mathsf{ApproxBal}(D, \alpha, \epsilon_0)$ that outputs $(1+\epsilon_0)$-approximate b_D in $O(m\alpha^2\epsilon_0^{-2}\mathrm{poly}(\log n))$ time.*

By simply calling the algorithm in Lemma 3.1, we obtain an $O(mb_D^2\mathrm{poly}(\log n))$ time algorithm for constructing a cut sparsifier for a digraph D. In this section we shall present an improved implementation by first showing the following.

Lemma 3.2. *Given a weighted digraph D, there is an algorithm that outputs α with $b_D \le \alpha \le 27b_D$ with probability at least $1 - 1/n^d$ in $O(m\log b_D + b_D^3 n \log W\,\mathrm{poly}(\log n))$ time, where W is the maximum weight of an edge in D.*

In the algorithm stated in Lemma 3.2, we use the following two algorithms as subroutines.

- $\mathsf{ApproxBal}(H, \alpha, \epsilon_0)$: Given $\alpha \in \mathbb{Z}_+, \epsilon_0 \in \mathbb{R}_+$, and an α-balanced digraph H with n vertices and m edges, output b with $b_H \le b \le (1+\epsilon_0)b_H$ in $O(m\alpha^2\epsilon_0^{-2}\,\mathrm{poly}(\log n))$ time.
- $\mathsf{Sparsify}(H, \alpha, \epsilon_0)$: Given $\alpha \in \mathbb{Z}_+, \epsilon_0 \in \mathbb{R}_+$, and an α-balanced digraph H with n vertices and m edges, output H' with $O(\alpha\epsilon_0^{-2} n \log n \log(nW))$ edges that $(1+\epsilon_0)$-cut-approximates H with probability at least $1 - 1/n^{d+1}$ in $O(m)$ time.

Note that in these subroutines we are required to know that the input is α-balanced in advance.

Combining these two subroutines, we consider Algorithm 2 to compute the imbalance approximately. Here D^{-1} denotes the digraph obtained from D by reversing the direction of each edge, and αD denotes the weighted digraph in which the weight of each edge is α times of the original weight.

We show that Algorithm 2 outputs a constant-factor-approximation of b_D. We first remark that, since there are at most $\log n$ iterations, all $\mathsf{Sparsify}(D_\alpha, \alpha, \epsilon_0)$ outputs a cut sparsifier with probability at least $1 - 1/n^d$.

Lemma 3.3. *For any $\alpha \in \mathbb{Z}_+$, $D_\alpha = D \cup \alpha D^{-1}$ satisfies*

$$b_{D_\alpha} = \frac{1+\alpha b_D}{\alpha + b_D} \le \alpha.$$

Proof. For any nonempty subset $U \subsetneq V$,

$$\frac{\delta^-(U; D_\alpha)}{\delta^+(U; D_\alpha)} = \frac{\delta^-(U; D) + \alpha\delta^+(U; D)}{\delta^+(U; D) + \alpha\delta^-(U; D)} = \frac{1+\alpha\beta(U)}{\beta(U)+\alpha}$$

where $\beta(U) := \delta^+(U; D)/\delta^-(U; D)$. For $a \ge 1$, a function $f(x) = (1+ax)/(a+x) = a - (a^2 - 1)/(a+x)$ is monotonically increasing. Hence, b_{D_α} is given by U that maximizes $\beta(U)$, implying the first equation in the statement.

Algorithm 2 An algorithm to approximate imbalance

Input: A weighted digraph $D = (V, E, w)$

$\epsilon_0 \leftarrow 0.1$, $\epsilon_1 \leftarrow 2(1+\epsilon_0)/(1-\epsilon_0)$, $\alpha \leftarrow 1$
while $\alpha \leq n$ **do**
  $D_\alpha \leftarrow D \cup \alpha D^{-1}$
  $H_\alpha \leftarrow \mathsf{Sparsify}(D_\alpha, \alpha, \epsilon_0)$
  $b_\alpha \leftarrow \mathsf{ApproxBal}(H_\alpha, \alpha(1+\epsilon_0)/(1-\epsilon_0), \epsilon_0)$
  if $b_\alpha \leq (\alpha + \alpha^{-1})/2\epsilon_1$ **then**
    return α
  else
    $\alpha \leftarrow 2\alpha$
  end if
end while
Output "α is larger than n"

The second inequality simply follows by observing

$$b_{D_\alpha} = \alpha - \frac{\alpha^2 - 1}{\alpha + b_D} \leq \alpha.$$

□

Lemma 3.4. *Let* $H_\alpha = \mathsf{Sparsify}(D_\alpha, \alpha, \epsilon_0)$ *and* $b_\alpha = \mathsf{ApproxBal}(H_\alpha, \alpha(1+\epsilon_0)/(1-\epsilon_0), \epsilon_0)$*. Then with probability at least* $1 - 1/n^{d+1}$*,*

$$\frac{1-\epsilon_0}{1+\epsilon_0} b_{D_\alpha} \leq b_\alpha \leq \frac{(1+\epsilon_0)^2}{1-\epsilon_0} b_{D_\alpha}.$$

Proof. From Lemma 3.3, $\mathsf{Sparsify}(D_\alpha, \alpha, \epsilon_0)$ correctly outputs a cut sparsifier with probability at least $1 - 1/n^{d+1}$. Hence,

$$\frac{1-\epsilon_0}{1+\epsilon_0} b_{D_\alpha} \leq b_{H_\alpha} \leq \frac{1+\epsilon_0}{1-\epsilon_0} b_{D_\alpha}.$$

By Lemma 3.3 this in particular implies $b_{H_\alpha} \leq \alpha(1+\epsilon_0)/(1-\epsilon_0)$, and therefore $\mathsf{ApproxBal}(H_\alpha, \alpha(1+\epsilon_0)/(1-\epsilon_0), \epsilon_0)$ correctly outputs a $(1+\epsilon_0)$-approximate of b_{H_α}, i.e., $b_{H_\alpha} \leq b_\alpha \leq (1+\epsilon_0) b_{H_\alpha}$. Therefore we obtain the relation in the statement. □

Lemma 3.5. *Let* $b_\alpha = \mathsf{ApproxBal}(H_\alpha, \alpha(1+\epsilon_0)/(1-\epsilon_0), \epsilon_0)$*, and suppose that* $b_\alpha \leq (\alpha + \alpha^{-1})/2\epsilon_1$ *where* $\epsilon_1 = 2(1+\epsilon_0)/(1-\epsilon_0)$*. Then* $\alpha \geq b_D$ *with probability at least* $1 - 1/n^{d+1}$*.*

Proof. If $b_\alpha \leq (\alpha + \alpha^{-1})/2\epsilon_1$,

$$\frac{1-\epsilon_0}{1+\epsilon_0} \cdot \frac{1 + \alpha b_D}{\alpha + b_D} \leq \frac{(1-\epsilon_0)}{2(1+\epsilon_0)} \cdot \frac{1}{2}\left(\alpha + \frac{1}{\alpha}\right) \tag{1}$$

holds from Lemma 3.3 and Lemma 3.4. Then (1) is equivalent to

$$0 \leq \alpha^3 - 3\alpha^2 b_D - 3\alpha + b_D = \alpha(\alpha - b_D)(\alpha - 1) - (2b_D - 1)\alpha^2 - b_D(\alpha - 1) - 3\alpha.$$

Since $\alpha \geq 1$ and $b_D \geq 1$, it is necessary that $\alpha \geq b_D$. □

Lemma 3.6. *Let* $b_\alpha = \mathsf{ApproxBal}(H_\alpha, \alpha(1+\epsilon_0)/(1-\epsilon_0), \epsilon_0)$ *and* $\Delta = 4\epsilon_1(1+\epsilon_0)^2/(1-\epsilon_0)$*. If* $\alpha \geq \Delta b_D$*, then* $b_\alpha \leq (\alpha + \alpha^{-1})/2\epsilon_1$ *with probability at least* $1 - 1/n^{d+1}$*.*

Proof.

$$\begin{aligned}
b_\alpha &\le \frac{(1+\epsilon_0)^2}{1-\epsilon_0} b_{D_\alpha} && \text{(by Lemma 3.4)} \\
&= \frac{(1+\epsilon_0)^2}{1-\epsilon_0} \cdot \frac{1+\alpha b_D}{\alpha + b_D} && \text{(by Lemma 3.3)} \\
&\le \frac{(1+\epsilon_0)^2}{1-\epsilon_0} \left(\frac{1}{\alpha + b_D} + \frac{\alpha}{\Delta} \right) && \text{(by } \alpha \ge \Delta b_D\text{)} \\
&\le \frac{(1+\epsilon_0)^2}{1-\epsilon_0} \cdot \frac{2\alpha}{\Delta} && \left(\text{by } \frac{1}{\alpha + b_D} \le \frac{1}{2} < b_D \le \frac{\alpha}{\Delta}\right) \\
&\le \frac{(1+\epsilon_0)^2}{1-\epsilon_0} \cdot \frac{2}{\Delta} \left(\alpha + \frac{1}{\alpha} \right) \\
&= \frac{1}{2\epsilon_1} \left(\alpha + \frac{1}{\alpha} \right)
\end{aligned}$$

□

We are now ready prove Lemma 3.2.

Proof of Lemma 3.2. Let α^* be the output. By Lemma 3.5 we have $b_D \le \alpha^*$. By Lemma 3.6 and Line 6 of Algorithm 2, it holds that α in the second to last loop ($= \alpha^*/2$) is at most Δb_D. Thus, by the definition of Δ and ϵ_1,

$$\alpha^* \le 2\Delta b_D = \frac{16(1+\epsilon_0)^3}{(1-\epsilon_0)^2}.$$

When we take $\epsilon_0 = 0.1$, we have $\alpha^* \le 27 b_D$.

The time complexity can be obtained by replacing m of the time complexity of ApproxBal with the edge size of the output of Sparsify. □

Now, by using Algorithm 2 to compute the imbalance of a given digraph, we have the following computational result for cut sparsifiers.

Theorem 3.7. *Given a weighted digraph D and ϵ, there is an algorithm that outputs a cut sparsifier with $O(b_D \epsilon^{-2} n \log n \log(nW))$ edges in expectation with probability at least $1 - 1/n^d$ in time $O(m \log b_D + b_D^3 n \log W \operatorname{poly}(\log n))$, where W is the maximum weight of an edge in D.*

4 Minimum Cut Problem

Ene et al. [5] show the following algorithm.

Theorem 4.1 (Ene et al. [5])**.** *Given a weighted digraph D, a source s, a sink t, and ϵ_0 with $0 < \epsilon_0 < 1$, there is an algorithm that outputs a $(1+\epsilon_0)$-approximate minimum s-t cut in time $O(m b_D^2 \epsilon_0^{-2} \operatorname{poly}(\log n))$.*

When computing a $(1+\epsilon_0)$-approximate minimum s-t cut, there is a simple trick to suppose that the edge weight is integer-valued and the maximum weight value is at most $O(m^2/\epsilon_0)$ [3]. Hence by using the cut sparsifier in Theorem 3.7 at a preprocessing phase in the algorithm in Theorem 4.1, we have the following.

Theorem 4.2. *Given a weighted digraph D, a source s, a sink t, and ϵ_0 with $0 < \epsilon_0 < 1$, there is an algorithm that outputs a $(1+\epsilon_0)$-approximate minimum s-t cut with probability at least $1 - 1/n^d$ in $O(m \log b_D + b_D^3 \epsilon_0^{-4} n \operatorname{poly}(\log n))$ time.*

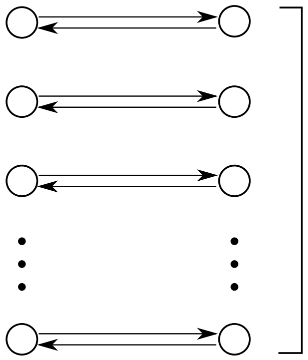

Figure 1: An example that shows the simple approach for counting does not seem to work.

5 Bound of the number of cut projections in balanced digraphs

It was shown by Karger and Stein [10] that the number of cuts of size at most β times the minimum cut size is bounded by $n^{2\beta}$ for any undirected graph with n vertices. This was generalized by Fung et al. [6] in the form of cut projections (defined below for digraphs), and was crucially used in the analysis of cut sparsifiers. On the other hand, there is a family of digraphs for which the number of the minimum cuts exponentially grows in n, and this is a critical difference between undirected and directed graphs. In view of this, in this section we shall give a new bound on the number of cut projections in terms of imbalance.

Definition 5.1 (Fung et al. [6])**.** *An edge is said to be* k-heavy *if its connectivity is at least k; otherwise, it is said to be* k-light*. The* k-projection *of an edge set is the set of k-heavy edges in it.*

The following theorem, a natural extension of the theorem by Fung et al. [6], is a key tool in the proof of Theorem 2.3.

Theorem 5.2. *Let λ be the weight of a minimum weight cut in a digraph D. Then, for any integer $k \geq \lambda$ and any real number $\beta \geq 1$, the number of k-projections of cuts with size at most βk is at most $2n^{2\beta b_D}$.*

Suppose that D is an Eulerian digraph, i.e., $b_D = 1$. One natural idea to prove Theorem 5.2 for D is to apply the undirected version of Fung et al. [6] to the underlying undirected graphs. Specifically for counting the number of cuts of size k, we may count the number of cuts of size $2k$ in the underlying undirected graph since in the Eulerian digraph $D = (V, E)$ we have $\delta^+(U; D) = \delta^-(U; D)$ for any $U \subseteq V$. This simple approach does not seem to work. Consider, for example, the graph in Fig. 1, which consists of n disjoint pairs of strongly connected graphs of two vertices. The corresponding undirected graph consists of n pairs of vertices with two multiple edges. Consider counting the number of cuts of size n in this digraph. This corresponds to picking one vertex from each component, and hence we have 2^n choices. On the other hand, in the underlying graph, there is only one cut of size $2n$, that is, the whole edge set.

We may guess such a gap does not occur in strongly connected digraphs; however we cannot ignore disconnected digraphs in order to count the number of k-projections stated in Theorem 5.2; even if a given digraph is strongly connected, it can be disconnected after removing k-light edges.

Nevertheless we can apply the proof of the undirected counterpart by Fung et al. [6]. One critical ingredient in the proof by Fung et al. [6] is Mader's splitting-off theorem, whose directed counterpart does not hold in general. Fortunately, Jackson [7] already pointed out an extension of Mader's theorem to Eulerian digraphs, and this extension enables us to apply the proof of Fung et al. [6] to Eulerian digraphs. Extending the result from Eulerian digraphs to general digraphs using the imbalance parameter is done by a simple counting argument. See Appendix A for a formal proof.

References

[1] András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 47–55, 1996.

[2] András A. Benczúr and David R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM Journal on Computing*, 44(2):290–319, 2015.

[3] Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng. Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 273–281, 2011.

[4] Michael B. Cohen, Jonathan Kelner, John Peebles, Richard Peng, Anup B. Rao, Aaron Sidford, and Adrian Vladu. Almost-linear-time algorithms for Markov chains and new spectral primitives for directed graphs. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, pages 410–419, 2017. A full version is available at `http://arxiv.org/abs/1611.00755`.

[5] Alina Ene, Gary Miller, Jakub Pachocki, and Aaron Sidford. Routing under balance. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, pages 598–611, 2016.

[6] Wai Shing Fung, Ramesh Hariharan, Nicholas J. A. Harvey, and Debmalya Panigrahi. A general framework for graph sparsification. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 71–80, 2011. A full version is available at `https://www.cs.ubc.ca/~nickhar/Publications/Sparsifier/Sparsifier-Long.pdf`.

[7] Bill Jackson. Some remarks on arc-connectivity, vertex splitting, and orientation in graphs and digraphs. *Journal of Graph Theory*, 12(3):429–436, 1988.

[8] Voratas Kachitvichyanukul and Bruce W. Schmeiser. Binomial random variate generation. *Communications of the ACM*, 31(2):216–222, 1988.

[9] David R. Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999.

[10] David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4):601–640, 1996.

[11] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: solving linear programs in $\tilde{O}(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow. In *Proceedings of IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 424–433, 2014.

[12] Hiroshi Nagamochi and Toshihide Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM Journal on Discrete Mathematics*, 5(1):54–64, 1992.

[13] Hiroshi Nagamochi and Toshihide Ibaraki. A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7(1):583–596, 1992.

[14] Richard Peng. Approximate undirected maximum flows in $O(m\,\text{polylog}(n))$ time. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1862–1867, 2016.

[15] Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

A Proof of Theorem 5.2

Let $D = (V, E)$ be a strongly connected digraph. Since the edge weight is integer-valued, in the following discussion we may assume that D is an unweighted multigraph. For $U \subseteq V$, we define $C^+(U; D)$ (resp., $C^-(U; D)$) be the set of edges from U to $V \setminus U$ (resp., from $V \setminus U$ to U). We also define $P(k, \beta; D)$ to be the set of the k-projections of cuts with size at most βk in D. Our goal is to prove $|P(k, \beta; D)| \leq 2n^{2\beta b_D}$.

We first consider the case when D is Eulerian.

Lemma A.1. *Let λ be the weight of a minimum weight cut in an Eulerian digraph D. Then, for any integer $k \geq \lambda$ and any real number $\beta \geq 1$, $|P(k, \beta, D)| \leq 2n^{2\beta}$.*

To prove this, we introduce the splitting-off operation.

Definition A.2. *The* splitting-off *operation replaces a pair of edges (u, v) and (v, w) with the edge (u, w), and is said to be* admissible *if it does not change the edge connectivity k_{st} between any two vertices $s, t \neq v$. It is well-known that splitting-off operation never increases the size of any cut.*

The complete splitting-off *operation at a vertex v repeatedly performs admissible splitting-off operations on the edges incident on v until v becomes an isolated vertex, and then removes v.*

Lemma A.3 (Jackson [7])**.** *Let v be a non isolated vertex of an Eulerian digraph D. Then, there exists a complete splitting-off operation at v.*

The proof of Lemma A.1 is done by analyzing the following algorithm, Algorithm 3, which is identical to that given by Fung et al. [6]. In Algorithm 3, a vertex v is said to be *k-heavy* if there exists an k-heavy edge incident on v; otherwise, it is said to be *k-light*.

Algorithm 3 performs a set of iterations. In each iteration, it performs complete splitting-off at all k-light vertices in D, contracts an edge selected uniformly at random, and removes all self-loops. The iterations terminate when at most $\lceil 2\beta \rceil$ vertices are left in the graph. At this point, the algorithm outputs the k-projection of a cut selected uniformly at random. Note that a complete splitting-off adds new edges to D. All new edges are treated as k-light irrespective of their connectivity. Therefore, the k-projection of a cut that is output by the algorithm does not include any new edge.

Lemma A.1 follows from the following.

Lemma A.4. *Let F be the k-projection of a cut with size at most βk. Then, Algorithm 3 outputs F with probability at least $n^{-2\beta}/2$.*

Indeed, if Lemma A.4 holds, the probability that a k-projection of a cut with size at most βk is returned is at least $n^{-2\beta}/2$ times the number of such k-projections. Thus we get $|P(k, \beta, D)| \leq 2n^{2\beta}$.

We shall now prove Lemma A.4. Algorithm 3 changes a graph to a different graph by complete splitting-offs, edge-contractions, and removals of self-loops. Let $D_i = (V_i, E_i)$ $(i = 0, \ldots, M)$ be the graphs which we consider during the algorithm. To prove Lemma A.4, we consider the following properties:

Algorithm 3 An algorithm for proving bound on cut projections

Input: An Eulerian digraph $D = (V, E)$, an integer $k \geq \lambda$ where λ is the weight of a minimum weight cut in D, and a real number $\beta \geq 1$

```
while there are more than ⌈2β⌉ vertices remaining do
    while there exists a k-light vertex v in D do
        Perform a complete splitting-off at v
    end while
    Pick an edge e uniformly at random
    Contract e and remove all self-loops
end while
return the k-projection of a cut selected uniformly at random
```

- (I1) D_i is Eulerian.
- (I2) There exists a subset $U_i \subseteq V_i$ such that $\mathrm{pr}_k(C^+(U_i; D_i); D) = F$ and $\delta^+(U_i; D_i) \leq \beta k$, where $\mathrm{pr}_k(C; D)$ is the k-projection of C in D.
- (I3) If $e \in E_i$ is not an edge added by complete splitting-offs and $\kappa(e; D) \geq k$, then $\kappa(e; D_i) \geq k$.

Clearly, $D = D_0$ has the properties (I1)-(I3). Since the removal of a self-loop does not affect any cut set, (I1)-(I3) are preserved. For a complete splitting-off operation,

- (I1) is preserved from the definition of splitting-off.
- (I2) is preserved since we only split-off at a k-light vertex and a splitting-off never increases the size of any cut.
- (I3) is preserved since we only split-off at a k-light vertex and the splitting-offs are admissible.

Lemma A.5. *Let D_{i+1} be the result of a contraction of an edge $f = (w, x)$ chosen from D_i uniformly at random. Suppose that D_i has the properties (I1)-(I3). Then, D_{i+1} has the properties (I1)-(I3) with probability at least $1 - 2\beta/|V_i|$.*

Proof. Clearly, (I1) is preserved. For (I3), since a contraction does not create new cuts, the edge connectivity of an uncontracted edge cannot decrease. Now we consider the probability that D_{i+1} has (I2). (I2) is preserved if $C^+(U_i; D_i) \cup C^-(U_i; D_i)$ does not contain f, and,

$$\begin{aligned}\Pr[f \notin C^+(U_i; D_i) \cup C^-(U_i; D_i)] &= 1 - \frac{|C^+(U_i; D_i) \cup C^-(U_i; D_i)|}{|E_i|} \\ &= 1 - \frac{2\delta^+(U_i; D_i)}{|E_i|}.\end{aligned}$$

Since every vertex in D_i is k-heavy, the outdegree of each vertex is at least k. Therefore, we have

$$|E_i| = \sum_{v \in V_i} \delta^+(v; D_i) \geq k|V_i|,$$

and

$$\Pr[f \notin C^+(U_i; D_i) \cup C^-(U_i; D_i)] \geq 1 - \frac{2\delta^+(U_i; D_i)}{k|V_i|} \geq 1 - \frac{2\beta}{|V_i|}.$$

□

We use a following technical lemma.

Lemma A.6 (Karger [9, p.42, ll.27–28])**.** *For any real number $\beta \geq 1$ and any positive integer $n > 2\beta$,*

$$\frac{n!}{\Gamma(n-2\beta+1)} < n^{2\beta}.$$

Now we are ready to prove Lemma A.4.

We assume that $n > 2\beta$; otherwise there is nothing to prove. Let N be the number of contractions, and suppose that the ith contraction transforms D_{j_i} into D_{j_i+1}. The output is F if $D_0, D_1, \ldots, D_M$ satisfies (I1)-(I3) and the algorithm selects the cut defined by U_M in Line 8. Let R be $\lceil 2\beta \rceil$. Then,

$$\begin{aligned}
&\Pr[\text{Algorithm 3 outputs } F] \\
&\geq \left(1-\frac{2\beta}{|V_{j_1}|}\right)\left(1-\frac{2\beta}{|V_{j_2}|}\right)\cdots\left(1-\frac{2\beta}{|V_{j_N}|}\right)2^{-|V_M|} \\
&= \left(1-\frac{2\beta}{n}\right)\left(1-\frac{2\beta}{n-1}\right)\cdots\left(1-\frac{2\beta}{R+1}\right)2^{-R} \\
&= \frac{n-2\beta}{n}\cdot\frac{n-1-2\beta}{n-1}\cdots\frac{R+1-2\beta}{R+1}\cdot 2^{-R} \\
&= \frac{\Gamma(n-2\beta+1)}{\Gamma(R-2\beta+1)}\cdot\frac{R!}{n!}\cdot 2^{-R} \geq \frac{\Gamma(n-2\beta+1)}{2\cdot n!} > n^{-2\beta}/2,
\end{aligned}$$

where the second last inequality follows from $2^{R-1} \leq R!$ and $0 < \Gamma(x) \leq 1$ for $1 \leq x \leq 2$, and the last inequality follows from Lemma A.6. This completes the proof of Lemma A.4, and hence Lemma A.1.

Proof of Theorem 5.2. Let $D' = D \cup D^{-1}$. For any $U \subseteq V$, it follows from the definition of imbalance that

$$(1+b_D^{-1})\delta^+(U;D) \leq \delta^+(U;D') \leq (1+b_D)\delta^+(U;D). \tag{2}$$

From the first inequality of (2), for any $u, v \in V$,

$$\begin{aligned}
\kappa((u,v);D') &= \min\left\{\delta^+(U;D') \mid U \subseteq V, u \in U, v \notin V\right\} \\
&\geq \min\left\{(1+b_D^{-1})\delta^+(U;D) \mid U \subseteq V, u \in U, v \notin V\right\} \\
&= (1+b_D^{-1})\kappa((u,v);D).
\end{aligned}$$

Hence for k-heavy edge e in D,

$$\kappa(e;D') \geq (1+b_D^{-1})\kappa(e;D) \geq (1+b_D^{-1})k.$$

Furthermore, from the second inequality of (2), if $U \subseteq V$ satisfies $\delta^+(U;D) \leq \beta k$, then $\delta^+(U;D') \leq \beta(1+b_D)k$. Thus,

$$\begin{aligned}
|P(k,\beta;D)| &= |\{\mathrm{pr}_k(C^+(U;D);D) \mid U \subseteq V,\ \delta^+(U;D) \leq \beta k\}| \\
&\leq |\{\mathrm{pr}_{(1+b_D^{-1})k}(C^+(U;D');D') \mid U \subseteq V,\ \delta^+(U;D') \leq \beta(1+b_D)k\}|.
\end{aligned}$$

Note that the last formula is $|P((1+b_D^{-1})k, \beta b_D; D')|$. Thus, by Lemma A.1, we have $|P(k,\beta;D)| \leq |P((1+b_D^{-1})k, \beta b_D; D')| \leq 2n^{2\beta b_D}$. ∎

B Proof of Theorem 2.3

The proof is again an adaptation of that for the undirected counterpart [6].

We prepare some notations. Recall that $C^+(U; D)$ is the set of edges from U to $V \setminus U$. For $U \subseteq V$, we define $F_i^{(U)} = F_i \cap C^+(U; D)$, and $f_i^{(U)} = |F_i^{(U)}|$. Let $\widehat{f_i^{(U)}}$ be the sum of weight over edges in $F_i^{(U)}$ that appear in D_ϵ. It holds that $\mathbb{E}[\widehat{f_i^{(U)}}] = f_i^{(U)}$.

The following Chernoff bound will be used.

Lemma B.1 (Fung et al. [6])**.** *Let $X_1, X_2, \ldots, X_n$ be n independent random variables such that X_i takes value $1/p_i$ with probability p_i and 0 otherwise. Then, for any p such that $p \le p_i$ for each i, any $\epsilon \in (0, 1)$, and any $N \ge n$,*

$$\Pr\left[\left|\sum_{i=1}^{n} X_i - n\right| > \epsilon N\right] < 2\mathrm{e}^{-0.38\epsilon^2 pN}.$$

The following lemma is a key to prove Theorem 2.3.

Lemma B.2. *Let $D_0, \ldots, D_\Lambda$ be a γ-certificate family of weighted Eulerian digraphs that covers D, and $i \in \{0, 1, \ldots, \Lambda\}$. Then with probability at least $1 - 1/n^{d+2}$, any $U \subseteq V$ satisfies*

$$|f_i^{(U)} - \widehat{f_i^{(U)}}| \le \frac{\epsilon}{2} \max\left\{\frac{\delta^+(U; D_i)}{\gamma}, f_i^{(U)}\right\}. \tag{3}$$

Proof. If $f_i^{(U)} = 0$, (3) holds with probability one. So we only consider U such that $f_i^{(U)} > 0$. By the connectivity condition of γ-certificates, we have $\delta^+(U; D_i) \ge 2^{i-1}$ for any such U. Then we partition subsets of V into $\mathcal{U}_{ij}$ $(j \ge 0)$ based on $\delta^+(U; D_i)$:

$$\mathcal{U}_{ij} = \{U \subseteq V \mid f_i^{(U)} > 0,\ 2^{i+j-1} \le \delta^+(U; D_i) \le 2^{i+j} - 1\}.$$

In order to analyze the worst situation, we may assume that each edge is sampled with probability strictly less than one, i.e, $p_e = \frac{\rho}{\lambda_e}$. We claim the following:

$$\text{Each } U \in \mathcal{U}_{ij} \text{ satisfies (3) with probability at least } 1 - 2n^{-(d+7)2^j}. \tag{4}$$

To see this, recall that $\lambda_e < 2^{i+1}$ for each $e \in F_i^{(U)}$. Hence

$$p_e = \frac{\rho}{\lambda_e} \ge \frac{\rho}{2^{i+1}}.$$

Therefore by Lemma B.1, we have

$$\begin{aligned}
&\Pr\left[|f_i^{(U)} - \widehat{f_i^{(U)}}| > \left(\frac{\epsilon}{2}\right) \max\left\{\frac{\delta^+(U; D_i)}{\gamma}, f_i^{(U)}\right\}\right] \\
&< 2\exp\left(-0.38\,\frac{\epsilon^2}{2^2}\,\frac{\rho}{2^{i+1}} \max\left\{\frac{\delta^+(U; D_i)}{\gamma}, f_i^{(U)}\right\}\right) \\
&\le 2\exp\left(-0.38\,\frac{\epsilon^2}{2^2}\,\frac{\rho}{2^{i+1}}\,\frac{\delta^+(U; D_i)}{\gamma}\right).
\end{aligned}$$

Using $\delta^+(U; D_i) \ge 2^{i+j-1}$ and $\rho = C\gamma \ln n/\epsilon^2$ with $C = 43(d+7)$, the last term is bounded by $2n^{-(d+7)2^j}$.

By (4) and the union bound, the failure probability of (3) is at most

$$\sum_{j \ge 0} |\{F_i^{(U)} \mid U \in \mathcal{U}_{ij}\}| \cdot 2n^{-(d+7)2^j}. \tag{5}$$

To bound $|\{F_i^{(U)} \mid U \in \mathcal{U}_{ij}\}|$ we use Theorem 5.2. By the connectivity condition of γ-certificates,

$$|\{F_i^{(U)} \mid U \in \mathcal{U}_{ij}\}| \le |\{F_i^{(U)} \mid \delta^+(U; D_i) \le 2^{i-1} \cdot 2^{j+1} - 1\}| \le |P(2^{i-1}, 2^{j+1}; D_i)| \le 2n^{4 \cdot 2^j}. \quad (6)$$

By (5) and (6) the failure probability of (3) is at most

$$\sum_{j \ge 0} 4n^{-(d+3)2^j} \le \frac{4n^{-(d+3)}}{1 - n^{-(d+3)}} \le \frac{1}{n^{d+2}}.$$

□

Proof of Theorem 2.3. In the graph D, there are at most n^2 pairs of vertices, so the number of distinct λ_e is at most n^2. Hence the number of nonempty F_i is at most n^2. Using union bound over these values of i, we can conclude that (3) is satisfied for all i and U with probability at least $1 - 1/n^d$. Thus, from the triangle inequality, we have

$$\begin{aligned}
|\delta^+(U; D) - \delta^+(U; D_\epsilon)| &= \left| \sum_{i=0}^{\Lambda} f_i^{(U)} - \sum_{i=0}^{\Lambda} \widehat{f_i^{(U)}} \right| \\
&\le \sum_{i=0}^{\Lambda} |f_i^{(U)} - \widehat{f_i^{(U)}}| \\
&\le \frac{\epsilon}{2} \sum_{i=0}^{\Lambda} \max \left\{ \frac{\delta^+(U; D_i)}{\gamma}, f_i^{(U)} \right\} \\
&\le \frac{\epsilon}{2} \left(\sum_{i=0}^{\Lambda} \frac{\delta^+(U; D_i)}{\gamma} + \sum_{i=0}^{\Lambda} f_i^{(U)} \right) \\
&\le \epsilon \cdot \delta^+(U; D)
\end{aligned}$$

since γ-overlapped property and

$$\sum_{i=0}^{\Lambda} f_i^{(U)} = \delta^+(U; D).$$

Hence we conclude that $D_\epsilon \in (1 \pm \epsilon) D$.

Finally, observe that the expected number of edges in D_ϵ is $\sum_e (1 - (1 - p_e)^{w_e}) \le \sum_e w_e p_e = O(\sum_e \rho w_e / \lambda_e)$. This completes the proof. □